Snort based collaborative intrusion detection system using blockchain in SDN

Ujjan, Raja Majid Ali; Pervez, Zeeshan; Dahal, Keshav

Published in:
Proceedings of the 2019 13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)

DOI:
10.1109/SKIMA47702.2019.8982413

Published: 06/02/2020

Document Version
Peer reviewed version

Snort Based Collaborative Intrusion Detection System Using Blockchain in SDN

Raja Majid Ali Ujjan*, Zeeshan Pervez, Keshav Dahal
*School of Computing, Engineering and Physical Sciences
University of the West of Scotland
Paisley, PA1 2BE, UK
{raja.vjjan, zeeshan.pervez, keshav.dahal}@uws.ac.uk

***Abstract*—Due to the rapid increment of the cyber attacks, intrusion detection system (IDS) is shifting towards collaborative approaches. There is a huge demand for securing larger networking environments for providing a safeguard against threats. In order to optimize the feasible detection performance, Collaborative Intrusion Detection Networks (CIDN) approaches have been adopted in practical scenarios, which enables a group of IDS nodes to mutually share and exchange mandatory information with each other, for example, IDS-signatures, attacks alarms. However, CIDN networks are distributed in nature, such networks still face plenty of implementation problems, especially, insider intruder can easily dominate any of security node and leave the entire security system vulnerable. To achieve the trust-based communication between each of IDS node, the recent advancement in blockchain applications is considered as a good fit to create trust-based communication in CIDN networks. This work converges CIDN network and blockchain in SDN context. Firstly, we investigated existing related work and highlighted challenges and research gap towards blockchain in CIDN networks. Secondly, we utilised three collaborated Snort IDS to receive the latest signature update from Ryu and then to securely share such signatures updates to all other Snort nodes within test-bed. Our work is motivated to detect seven types of common attacks with collaborated signature-based IDS, which feasibly processes more packets to achieve satisfactory detection results. Overall the evaluation results show that with the adoption of blockchain protocols, the proposed CIDN network achieves 96% of TP rate detection rate for TCP, UDP and ICMP packets.**

***Index Terms*—Software Defined Networks, Open vSwitch, Snort, Blockchain, , Collaborative Intrusion Detection Networks (CIDN).**

I. Introduction

In Software Defined Networking (SDN) environment, building a customised network architecture is a game changer, where, we can flexibly transfer legacy network infrastructure to an innovative, open source and programmable infrastructure. The authors of [1] depicted that most of the current public and private networks are exponentially increasing due to our busy online life, where network complexities and issues are also arising during implementation. With a significant increment of cyber attackers, most of the single IDS detection applications have failed to accurately discover attacks in large networks, IDS collect limited information about attacks when these are deployed against heavy traffic [2]. Most of the legacy IDS based implementations could easily be bypassed by complex attacks such as Denial of Services (DoS), Slowris and by highly experienced cybercriminals. To maximise the performance pragmatically, Collaborative intrusion Detection Networks(CIDN) have been deployed, in which group of IDSs collectively achieve the huge amount of data from other nodes [3]. Although, IDS nodes are capable to share their signature-rules with each deployed nodes of the system, in order to improve detection efficiency and reducing false alarms [4], [5], [6]. However, such IDS deployment can become vulnerable for insider attack, due to the fact of interconnected and distributed nature. If any of that node is severely infected then it can lower down the detection performance at other nodes [7]. To deal with this issue, there is significant demand to create an effective security mechanism to provide potential safeguard to signature sharing rules within CIDN networks.

With the motivation of impact and adoption of Bitcoin, blockchain technologies and IDS based security systems are being widely attracted towards industries and academia, this system enables trusted individuals to easily connect other potential network entities with verifiable approach without the interception of any complex centralised application [8]. By utilising consensus approaches, the blockchain helps to provide transparent and protected data storage, in which the majority of stored data blocks can not be modified unless to update all sub-blocks. This approach in the blockchain is significantly required to share IDS signature rules in a protected way for various CIDN platforms.

From the recent development of blockchain applications, we are motivated to propose a collaborated signature-based IDS model, which implies blockchain application to securely share Snort signature rules with all integrated Snort nodes. Our work mainly focuses to apply blockchain for deploying a trusted Snort signature database with the help of SDN Ryu [9]. This idea ensures verified signature rules from the relevant domain, which enables collaborative CIDN to improve detection accuracy with heavy and unknown malicious traffic, this also helps to lower down manipulation and management efforts. Major contributions of our work are provided below:

978-1-7281-2741-5/19/$31.00 ©2019 IEEE

- To process more packets with collaborative Snort nodes for the purpose of higher detection accuracy in seven major attacks types.
- To create a trusted Snort signature rules sharing channel in Ryu and reducing HIDS influencing burden in CIDN. We utilised blockchain implementation with distributed Snort IDSs in SDN context, SDN Ryu controller only updates NIDS Snort signature rule then all integrated HIDS Snort nodes are automatically updated by the trusted channel of blockchain.
- For evaluation, we carried out two different experiments, in the first experiment, we investigated the performance of the proposed system with IDS based attack detection. In the second, experiment, we investigated the proposed system accuracy with packet-drop and packet-processing with various input traffic intensity by using Ryu and Snort integration in a virtual machine.

The rest of this paper is organised as: Section II introduces the background and related work. The Section III depicts main implementation and design. Section IV evaluates proposed model results. Finally, we conclude our work in Section V with future directions.

II. Background Related Work

In legacy networks, stand-alone IDS carries less information for implemented networks, where it tries to provide a potential safeguard against unknown malicious activities, this enables the system to become vulnerable for potential attackers. For example, a cyber attacker can initiate a complex attack like Distributed Denial of Services (DDoS) to compromise a single Intrusion Detection System (IDS), due to the fact that it is less able to acquire overall network traffic status. To resolve this IDS issue, there is an alarming need to provide a collaborative IDS based solution to optimize detection ability [3].

The existing literature widely represents distributed monitoring based work for decades. Some of the existing systems such as Distributed Intrusion Detection System (DIDS) [10], which was proposed in late 1991, this system was capable to monitor distributed heterogeneous computer networks. Addition to this, Event Monitoring Enabling Responses to Anomalous Live Disturbances (EMERALD) [11] was also proposed in 1991, this system was capable to record various intrusion activities within an abstract layer of a big network. This model was able to merge high volume events from legacy distributed IDS.

1) Collaborative intrusion detection (CIDN): For feasible detection performance, a CIDN system utilises IDS nodes to share its mandatory information to every collaborated node. The authors of [12] in 2006 discussed that each IDS node centrally dependant in CIDN networks. For motivation of this issue, they designed new CIDN systems based on decentralised locations and routing infrastructure. Moreover, their system assumes a trust-based peer system, which could easily be targeted by inside intruders. In collaborative intrusion detection systems, attacks from inside can easily create the biggest loss, from which point an intruder can easily consume resources and become dominant in that system.

In order to provide a safeguard against inside attackers, a CIDN system more likely requires an appropriate and feasible trust-based solution, in which each IDS nodes can ensure trust factors to improve reputation amongst all IDSs. The authors of [13] proposed an IDSs system, which was created on the basis of the P2P overlay. The trust-aware engine is utilised to correlate triggers and addition to this it also correlates adaptive scheme for IDS trust management. Similarly, the authors of [14] designed a game theory model for analysing the processes of P2P network reporting. They investigated that if system reputation is not compatible then a number of system peers nodes results in false malicious reporting.

Based on the aforementioned facts, the other authors of [15] also depicted CIDN model, in which IDS nodes trustworthiness is dependant on received answers, firstly they depicted a Host-based IDS (HIDS) framework, where each HIDS node achieve trustworthiness following by its CIDN experience. To improve this model effectiveness, the authors in [16] have differentiated that various IDSs may vary for detection purpose.

2) Blockchain based IDS: Most of the industries and many researchers are focusing to deploy blockchains based applications since last six years, most of these applications are utilised for specific use-cases with unique and specific blockchains requirements with unique and customised characteristics. For example, in existing literature, there are large number of cases, which are developed on the bases of improving privacy with different parties in the system, the major aim was to focus for storing data completely private [17]. Similarly, more implementations use identity schemes with cryptography application, this mechanism caters full unavailability and anonymity between different transactions [16]. Based on these versatile approaches, most industries and researchers have been trying to investigate blockchain based application potential.

The blockchain based technologies are widely used to store information in a decentralised way to provide safeguard against any deviations. It is also very vital to find out the feasible way how to deploy the blockchain system in intrusion detection systems. Most of the current research is investigating this gap, the authors of [18], have provided a blockchain based CIDN framework, in this method, set of false alarms from IDSs are considered as blockchain transactions. Then, most of collaborating IDSs nodes utilised unique communication protocol for stable transaction connection before generating them in a specific block. This approach can be used as a safeguard against intrusion in blockchains. However, the authors did not provide a systematic implementation in detail, addition to this they did not evaluate results with practical implementation. To deal with this issue, authors in [19] utilised some early insights based on the interception of IDSs and blockchains units, they also addressed some issues and challenges in this platform. According to the authors, blockchains can provide a positive impact on a distributed intrusion detection system with regard to data sharing, exchange of alarms and trust

based processing. Similarly, authors of [20] also introduced a CIoTA framework to utilised the blockchain mechanism for collaborative anomaly detection, but there were limited resources. Moreover, IDS agents can also help to protect blockchain applications.

In contrast, there are some works , which have been published with OpenFlow based protocol, such as authors of [21] discussed a ChainGuard, which is developed on OpenFlow based firewall, this approach is feasibly created for securing SDN based blockchains system. In this method, OpenFlow based switch traffic is propagated to the blockchain based ChainGuard node. The main purpose of this system is to lower down the unknown malicious activities from participating system nodes. Moreover, the authors of [22] also developed an effective DistBlockNet system, also known as SDN based distributed system for IoT security, in which blockchains were utilised as key elements. This mechanism enables other nodes interaction without the interception of any trust-based central controller.

III. Design of Snort based CIDN Network with blockchain

In this section, we depict major SDN based deployment and implementation. Our proposed model comprises blockchain based collaborative intrusion detection network, which uses Snort IDS as Host-based (HIDS) and Network-based (NIDS) with Ryu programmable flexibility. The major aim of the research is to provide a safeguard against insider attacks and improve attack detection accuracy by utilising collaborative Snort node with blockchains certificates between control-plane Ryu application and all Snort nodes signature database. We deployed *Ethereum Geth* to create Ethereum environment in Ubuntu 16.04 LTS, This environment enables to create genesis block with major digital transaction signatures, we combined and connected all transaction ID as a unique hash function. Each Snort node utilizes this hash value and newly processed transaction ID. Similarly, previous hash value and transaction ID is to be used as a new hash value for the next Snort node in the new block of the chain. However each Snort IDS node in blockchain links with its previous block via hash values, which results in a chain which is directly connected to unique genesis block. This procedure provides secure communication to all Snort IDS to accept rule-set from control-plane.

In VM-2, *Ethereum Geth* and Snort nodes were deployed, each node constitutes secure SSH key pair to establish the connection. We utilize python class for employing blockchain, where SHA256 hash function is utilized for encryption.

The Fig. 2 depicts the blockchain network between collaborative CIDN nodes, where Snort-1, Snort-2 and Snort-3 node communicate each other by using a hash function such as a public key and a private key pairs. In our proposed CIDN network, Snort-1 node connects to Snort-2 node via previous hash value, similarly, Snort-2 node connects to Snort-3 node. In this way, every node of Snort invites each other by the signed transaction. Moreover, after establishing secure communication, these nodes start to share Snort rule-set with each other with appropriate block key pairs.

TABLE I
Seven Different Attack types.

No	Attack-types	Description
1	DDoS-attacks	DDoS Floods (Metaslploit)
2	SSH-attacks	SSH Exploits (Metaslploit)
3	FTP-attacks	Brute Force (Metaslploit)
4	HTTP-attacks	HTTP Floods (Kali Linux)
5	ICMP-attacks	Smurf Attacks (Metaslploit)
6	ARP-attacks	ARP Spoofing (Kali Linux)
7	Scan-attacks	Port Scans (Kali Linux)

The proposed system detects unknown DDoS, including seven different types of attacks mentioned in TABLE I. This system helps to keep protected from insider attackers. In SDN architecture, IDS nodes are integrated with blockchain trusted channel which enables SDN controller to effectively share signatures rules to Snort nodes, which are implemented in data-plane. Our design systematically operates as a loop of three components such as Ryu controller, IDS nodes and blockchain trust certificates, as depicted in Fig.1. The IDS depicts the DDoS attack detection mechanism, data-plane represents the network in which different Snort nodes are integrated by blockchain to share signature automatically. However, SDN Ryu controller plays a very vital role to manipulate and update the blockchain hash table and all other Snort and OpenVswith entries.

To evaluate this work, we carried out two different experiments with three virtual machines, which are created with Ubuntu LTS 16.04 64 bit OS. VM-1 machine is created with SDN Ryu and Snort integration. SDN controller primarily programmes the network operation such as updating Snort signatures rules and data-plane flow. This programme is initiated once Ryu receives the Packet_in message from OpenFlow. The Ryu controller changes the data-plane of proposed CIDN network by utilising Link-A, in this virtual machine, Link-B is used for Snort signature sharing towards CIDN network. VM-2 machine represents the network domain, where a network emulator (eg. Mininet) is utilised to create CIDN virtual network with Mininet simulator. CIDN network comprises of three Snort IDS nodes [23], all nodes were deployed with default signature rules. When a new packet _in arrives with malicious attributes then main HIDS Snort node receives new signature updates from Ryu via Link-D, then all other CIDN nodes are also updated via blockchain. SDN based switches use the OpenFlow protocol to communicate the VM-1 and VM-2. Once all Snort nodes from CIDN receives packets, then integrated switches use a port mirror approach for sending entire traffic streams to Link-B. We carried two experiments with CIDN with 100 Mbps data intensity to validate our model performance. This machine also utilises NAT node as a gateway so that some hosts can launch attacks as a malicious injection. VM-3 is used for launching attacks towards CIDN-1 and CIDN-2 nodes. This virtual machine acts as a Host-Only adapter.

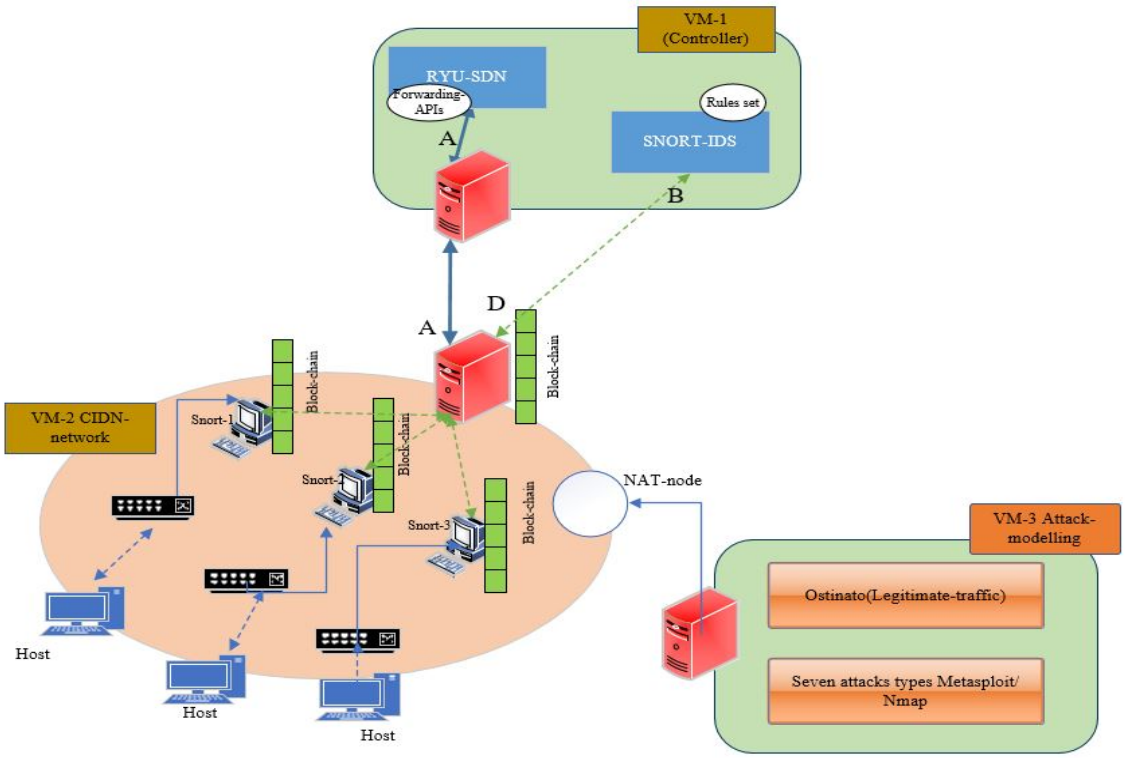

Fig. 1. Proposed Test-Bed of Snort IDS with DNN Co-Detection in SDN.

IV. Evaluation

In this section, we evaluate the performance of our proposed system in SDN based environment, where we use CIDN network to calculate the collaborative based Snort CIDN network performance with the implementation of blockchain. In our proposed model CIDN, uses blockchain based Snort nodes, which can monitor, identify unknown malicious activity and also share Snort signature rule-set with neighbouring nodes. Each node needs to sign privately for qualifying rule-set, each remaining node has to follow this rule-set. In this way, blockchain nodes expanded when these nodes are verified then these blocks receives the trusted rules only.

A. Attack modelling

VM-3 is used for attack modelling as a server, in which we are running HTTP, FTP and SSH services. We utilised the Metasploit framework and Kali Linux [24] in order to generate seven types of malicious traffic as shown in TABLE I. The purpose of using Metasploit framework is to generate malicious traffic with different payloads and exploitation for various operating systems such as Windows, Linux or Mac OS. All seven major attacks along with legitimate traffic injected to CIDN IDS nodes. All blockchain based IDS starts to inspect malicious and legitimate traffic and trigger alarms if the input traffic matches the rule set which is shared via blockchains to Snort IDS. These number of alarms (comprises as false-positive, false-negative and true-positive) will classify network traffic.

B. Evaluation with malicious and legitimate traffic in CIDN network

In this experiment, we investigated the performance of collaborated Snort IDS with blockchain based CIDN network. CIDN network is deployed to investigate performance between malicious and legitimate traffic. We have utilised three collaborated Snort IDS nodes with blockchain, each node is also integrated with SDN based switch. In this experiment, we have launched seven different types of malicious attacks into CIDN network, where each node is utilising Snort signature rule-set. We utilised only 3 nodes in CIDN network, in order to measure the CIDN detection accuracy we have utilised the following IDS performance metrics.

1) True Positive (TP) - Values to correctly identifying as attacks and non-attacks records.
2) True Negative (TN) - Values to correctly identifying only non attack values.
3) False Positive (FP) - Values to incorrectly predicting attack records.
4) False Negative (FN) - Values to incorrectly identifying non attacks.

$$TPrate = \frac{TP}{TP + FN} \tag{1}$$

$$FPrate = \frac{FP}{FP + TN} \tag{2}$$

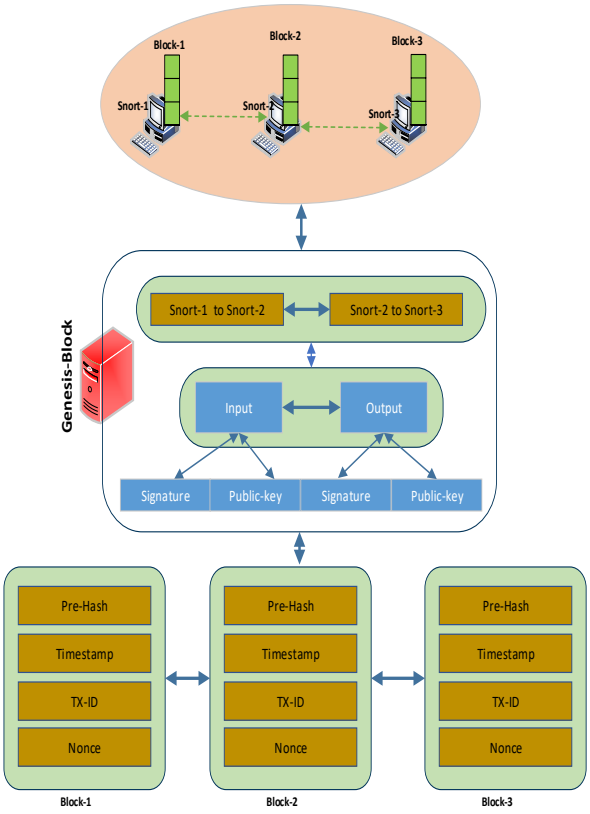

Fig. 2. Blockchain trust communication inside CIDN Snort nodes.

$$FNrate = \frac{FN}{FN + TP} \tag{3}$$

In CIDN, collaborated Snort IDS nodes were configured with the default rule set. However, We run our proposed test-bed for around two hours as shown in TABLE IV. The seven different attack types were injected with different speed to CIDN. In the first test, collaborated Snort IDS with default rule set performance was not acceptable. This is due to the fact that Snort is rule-based, it mainly focuses on rules set once input traffic attack matches then Snort generate an alarm. In this work, we are utilising seven attack combination with legitimate traffic. Snort collaborated IDSs of CIDN with a default rule set, only detected average combined TP rate 73.5% , 74% and 76% for Snort-node-1, Snort-node-2 and Snort-node-3 respectively. In the second run time, we enable collaborated IDS to receive rule set from Ryu controller by utilising trusted blockchain. As shown in TABLE II , we have depicted the three Snort collaborated nodes values, which are integrated with blockchain rules set sharing approach. The Snort-node-1 achieved average FP rate of 10% with six major attacks types, in this node, Snort-FN rate was recorded nearly 6%. In Snort-node-2 and Snort-node-3, ARP-attack, FTP-attack, ICMP-attack and FTP-attack achieved average of 10% FP rate. The average detection rate of true positive rate was calculated around 97% for each of six malicious traffic with Snort-node-1, Snort-node-2, Snort-node-3. However, Snort-node-1 only achieved 44% TP rate, Snort-node-2 achieved 44% TP rate and Snort-node-3 achieved less than 51% TP rate with Scan malicious traffic.

Fig. 3 depicts the average performance of Snort IDS nodes TP rate with respect to various traffic input intensity with Mbps. The overall detection performance of the proposed test-bed is evaluated with background running of the blockchain mechanism, which enables each IDS node to adopt new signatures from Ryu, which helps to detect more accurately and effectively. When the proposed system processes the 10 Mbps of combined legitimate and malicious traffic, the average TP detection rate of six attacks was stood more than 60%, where SSH-attacks and Scan-attacks were recorded with 74% of TP rate and 44% of TP rate respectively. Scan-attacks TP rate was not good and overall it was recorded up to 50% of TP rate. Average TP rate of DDoS-attacks,FTP-attacks, HTTP-attacks, ICMP-attacks and ARP-attacks were recorded more than 80% of TP between 30 Mbps to 60 Mbps traffic intensity. Similarly, all of these six attacks types achieved more than 90% of TP rate with 70 Mbps to 100 Mbps combined input traffic intensity. More ever, our proposed work achieved an average of 96% of TP detection rate once we increase traffic intensity up to 100Mbps, this detection accuracy was achieved within six major common attacks types only detail is depicted in Fig. 3.

C. Evaluation with legitimate traffic in CIDN network

In this experiment, we have utilised same CIDN network, which also comprises the same three Snort IDS, these nodes are collaborated with blockchain. To validate the proposed design, we observed the collaborated Snort IDS performance with the help of legitimate traffic. We generated legitimate traffic by utilising (Ostinato tool). We performed an experiment to investigate the performance of all collaborated Snort nodes with a network speed of 100Mbps. We injected 1,250 bytes of TCP, UDP and ICMP packets. As shown in Fig.1, we run Snort IDS nodes individually on the proposed test-bed. In this CIDN network, we have utilised a number of tools to record and investigate CPU utilisation, memory/network utilisation, and packet drop rate. The tools include dstat, Snort Barnyard2 log-file, TCP-dump, nmap and Metasploit framework etc. In this experiment, we manually injected packets with 300packets/sec to all nodes. Background network link speed is divided into the range of 1 to 100 Mbps. All collaborated Snort IDS were investigated with total accumulated packets of 10,800,000 TCP with 300 packets/sec intensity. Similarly, total accumulated packets of 10,800,000 of UDP and total accumulated packets of 10,800,000 of ICMP were also injected to CIDN network nodes in order to validate. The TCP, UDP and ICMP packets were manually injected with 300 packets/sec rate. The CPU-utilisation and memory-utilisation of CIDN network with blockchain based collaborated Snort IDS is depicted in the TABLE III.

TABLE II
DIFFERENT ATTACKS DETECTION ACCURACY WITH BLOCKCHAIN BASED CIDN.

Malicious-traffic	Snort-node-1		Snort-node-2		Snort-node-3	
	Snort-FP rate	Snort-FN rate	Snort-FP rate	Snort-FN rate	Snort-FP rate	Snort-FN rate
SSH-attacks	9.0%	2.0%	8.0%	4.0%	11.0%	2.0%
DDoS/DoS-attacks	5.0%	4.0%	6.5%	2.0%	8.0%	4.0%
FTP-attacks	10.0%	7.0%	9.0%	8.0%	14.5%	4.0%
HTTP-attacks	4.0%	8.0%	6.0%	3.0%	8.0%	2.0%
ICMP-attacks	13.0%	4.0%	9.0%	6.0%	11.5%	4.0%
ARP-attacks	12.0%	5.0%	11.0%	9.0%	12.5%	3.0%
Scan-attacks	44.0%	19.0%	50.0%	17.0%	44.5%	21.0%

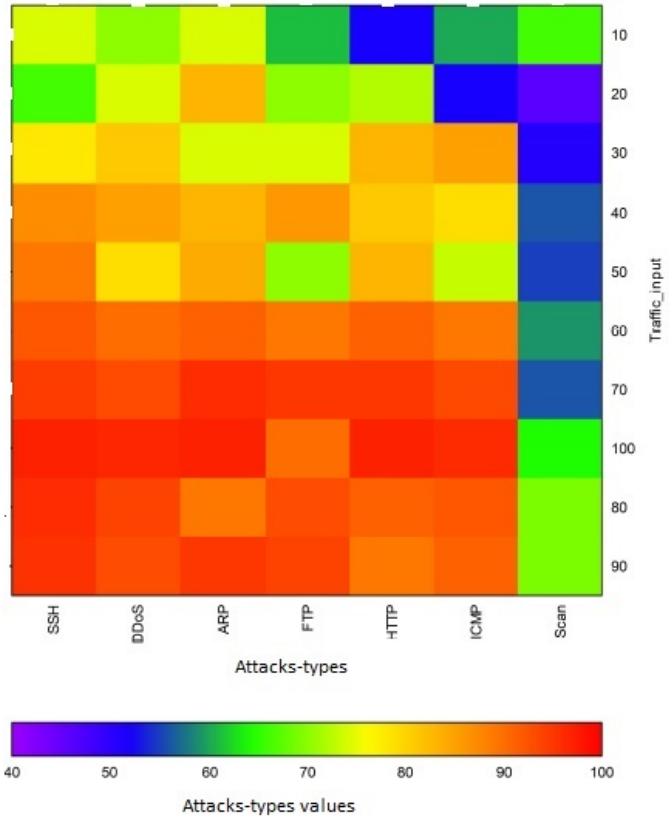

Fig. 3. CIDN network different attacks TP-rate with different traffic intensity.

The CPU utilisation of CIDN network with blockchain based collaborated Snort IDS is calculated in such a way as CIDN network contains three Snort IDSs nodes (eg Snort-node-1, Snort-node-2 and Snort-node-3) each of that nodes utilises shared processor unit, not overall CPU. The CPU consumption of all three nodes is investigated with Intel (R) Xeon (R) X5560 CPU with 2.88 GHz processor and 16 GB RAM (DDR3 ECC-Registered Memory PC3-12800MHZ).

From the collected data, we observed that all collaborated Snort IDS nodes were utilising almost the same CPU and memory. Snort-node-1 memory and CPU utilisation were almost identical with Snort-node-2 and Snort-node-3. From the TABLE III, it can be seen that each node CPU-utilisation is increasing due to the increment of input traffic intensity. The average CPU-utilisation of Snort IDS node is 65% when these nodes were receiving input traffic intensity 10 Mbps to 40 Mbps, then CPU exponentially increase nearly 80% CPU utilisation in all three collaborated Snort nodes. Each Snort node utilises only shared 2-cores CPU or 4-core CPU out of 8-core CPU.

In this experiment, each Snort node is able to classify nearly 2 Mbps out of 10 Mbps. With 20 Mbps and 30 Mbps input traffic intensity, all three nodes were able to inspect only 4 Mbps traffic. When we injected 80 Mbps,90 Mbps and 100 Mbps traffic burst then Snort-node-1 only processed 11 Mbps, Snort-node-2 only processed 13 Mbps and Snort-node-3 only processed 10 Mbps respectively. These calculation were for individual Snort IDS.

Moreover, Snort IDS is signature based and its processing is limited if used as stand-alone IDS during heavy network traffic burst. Once we inject 10 Mbps input traffic to CIDN based network then all nodes with the help of blockchain can easily process all input traffic. This enables CIDN network to correctly identify intrusions with very less false triggers. Blockchain based approach also helps Snort IDS to accept new signatures from Ryu controller.

When input network traffic intensity started to increase, the CPU and memory consumption also started to increase, this results in packets drop as well. We injected almost total accumulated packets of 10,800,000 for UDP, TCP and ICMP with 300 packets/sec intensity. From TABLE IV, we can observe that Snort IDS nodes with blockchain approach can process 20,000 UDP, 21,000 TCP and 23,000 ICMP packets with the 10 Mbps traffic intensity, TCP-packets drop rate was higher at this stage such as 7.0%. Once we increase the input traffic intensity up to 50% Mbps then the proposed CIDN test-bed processed an average of 58,000 of UDP, TCP and ICMP packets with the packet-drop rate of around 8%. We continued to double up input network traffic such as the 100 Mbps intensity of malicious and legitimate traffic then 63,000 of total UDP, TCP and ICMP packets were processed with blockchain-based IDS nodes, the packet drop rate was higher such as 21% of the total injected packets. Overall, we can observe that after combining all Snort nodes, the packet processing of the proposed system is feasibly improved to catch various attacks types rapidly.

TABLE III
DIFFERENT CPU AND MEMORY UTILISATION WITH BLOCKCHAIN BASED CIDN.

Traffic-intensity	Snort-node-1		Snort-node-2		Snort-node-3	
	CPU-utilisation	Memory-utilisation	CPU-utilisation	Memory-utilisation	CPU-utilisation	Memory-utilisation
10 Mbps	64%	3.0 Mbps	61%	2.0 Mbps	60%	1.2 Mbps
20 Mbps	66%	3.5 Mbps	64%	3.3 Mbps	62%	3.0 Mbps
30 Mbps	67%	5 Mbps	65%	4.0 Mbps	65%	4.5 Mbps
40 Mbps	68%	6.6 Mbps	69%	6.0 Mbps	66%	6.0 Mbps
50 Mbps	71%	8.9 Mbps	70%	9.0 Mbps	69%	7.0 Mbps
60 Mbps	71%	9 Mbps	72%	9.1 Mbps	71%	8.1 Mbps
70 Mbps	74%	10 Mbps	74%	11.0 Mbps	72%	9.0 Mbps
80 Mbps	76.0%	10.9 Mbps	75%	12.0 Mbps	74%	9.11 Mbps
90 Mbps	77.0%	11 Mbps	76%	12. 2 Mbps	76%	10.0 Mbps
100 Mbps	81.0%	12.3 Mbps	78%	12.9 Mbps	77%	10.1 Mbps

TABLE IV
DIFFERENT PACKET-PROCESSING AND PACKET-DROPS WITH BLOCKCHAIN BASED CIDN.

Traffic-input		UDP-packets		TCP-packets		ICMP-packets	
Bandwidth	time-elapsed	Packet-processing	packet-drops	Packet-processing	packet-drops	Packet-processing	packet-drops
10 Mbps	450	20,000	5.3%	21,000	7.0 %	23,000	1.2 %
20 Mbps	900	30,000	6.0 %	32,000	6.3 %	28,000	6.0 %
30 Mbps	1,350	40,000	6.8 %	40,000	4.0 %	34,000	5.5 %
40 Mbps	1,800	55,000	7.6 %	57,000	5.0 %	50,000	6.0 %
50 Mbps	2,250	58,000	8.4 %	56,000	9.0 %	58,000	8.0 %
60 Mbps	2,700	58,000	9 %	57,000	11.1 %	60,000	11.1 %
70 Mbps	3,150	61,000	11.5 %	62,000	12.0 %	60,000	9.0 %
80 Mbps	3,600	62,000	13.9 %	63,000	15.0 %	63,000	13.11 %
90 Mbps	4,050	63,000	15 %	60,000	19. 2 %	64,000	16.0 %
100 Mbps	4,500	65,000	17.3%	63,000	21.9 %	60,000	19.1 %

V. CONCLUSION

Collaborated intrusion detection approaches in SDN have received much attention for providing an effective safeguard against various malicious attacks in a larger network, which enables various IDS nodes to mutually and effectively share important information with each other such as, signature rules. However, if any of the collaborated nodes are infected then it shares untruthful Snort IDS signatures to all other IDS nodes which degrade the CIDN network performance. From the literature, blockchain technology is widely considered for providing verifiable information sharing approach without using any complex trust-based mechanism. With the motivation of current blockchain applications, our proposed work mainly focuses to utilise Snort signature-based detection and deploy SDN based test-bed in which we utilised three collaborated Snort IDS, which securely receive new signature updates from SDN Ryu controller. This collaborated approach of CIDN network with Snort IDS generates very less false alerts such as the average of 5% 0f FP rate and FN rate for DDoS and HTTP attacks. However, the average FP rate and FN rate for all other different attacks types were stood nearly 10%. Due to the fact that each collaborated IDS individually detects burst attack input traffic. This way IDS nodes process more packets as compared to stand alone IDS. In future work, we will deploy SDN based application for building trusted collaborated IDS framework via blockchain for scale-able and distributed network, we will utilize a comprehensive and effective mechanism with the unsupervised deep learning.

REFERENCES

[1] D. Kreutz, F. M. V. Ramos, P. E. Verssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan 2015.

[2] F. Gong, "Next generation intrusion detection systems (ids) mcafee network security technologies group," *McAfe Security*, apr 2003.

[3] Y.-S. Wu, B. Foo, Y. Mei, and S. Bagchi, "Collaborative intrusion detection system (cids): A framework for accurate and efficient ids," in *Proceedings of the 19th Annual Computer Security Applications Conference*, ser. ACSAC '03. Washington, DC, USA: IEEE Computer Society, 2003, pp. 234–. [Online]. Available: http://dl.acm.org/citation.cfm?id=956415.956425

[4] Y. Meng and L. for Kwok, "Enhancing false alarm reduction using voted ensemble selection in intrusion detection," *Int. J. Comput. Intell. Syst.*, vol. 6, pp. 626–638, 2013.

[5] Y. Meng, W. Li, and L.-f. Kwok, "Intelligent alarm filter using knowledge-based alert verification in network intrusion detection," in *Foundations of Intelligent Systems*, L. Chen, A. Felfernig, J. Liu, and Z. W. Raś, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 115–124.

[6] W. Li, W. Meng, C. Su, and L. F. Kwok, "Towards false alarm reduction using fuzzy if-then rules for medical cyber physical systems," *IEEE Access*, vol. 6, pp. 6530–6539, 2018.

[7] J. R. Douceur, "The sybil attack," in *Peer-to-Peer Systems*, P. Druschel, F. Kaashoek, and A. Rowstron, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 251–260.

[8] X. Xu, C. Pautasso, L. Zhu, V. Gramoli, A. Ponomarev, A. B. Tran, and S. Chen, "The blockchain as a software connector," in *2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA)*, April 2016, pp. 182–191.

[9] (2018, may) Ryu. available online:. Https://osrg.github.io/ryu/.

[10] J. Brentano, G. V. Dias, T. L. Goan, L. Todd Heberlein, C.-l. Ho, K. Levitt, B. Mukherjee, S. E. Smaha, T. Grance, D. M. Teal, and D. Mansur, "Dids (distributed intrusion detection system) - motivation, architecture, and an early prototype," 01 1999.

[11] P. A. Porras and P. G. Neumann, "Emerald: Event monitoring enabling responses to anomalous live disturbances," in *In Proceedings of the 20th National Information Systems Security Conference*, 1997, pp. 353–365.
[12] Z. Li, Y. Chen, and A. Beach, "Towards scalable and robust distributed intrusion alert fusion with good load balancing," in *Proceedings of the 2006 SIGCOMM Workshop on Large-scale Attack Defense*, ser. LSAD '06. New York, NY, USA: ACM, 2006, pp. 115–122. [Online]. Available: http://doi.acm.org/10.1145/1162666.1162669
[13] C. Duma, M. Karresand, N. Shahmehri, and G. Caronni, "A trust-aware, p2p-based overlay for intrusion detection," in *17th International Workshop on Database and Expert Systems Applications (DEXA'06)*, Sep. 2006, pp. 692–697.
[14] T. A. Tuan, "A game-theoretic analysis of trust management in p2p systems," in *2006 First International Conference on Communications and Electronics*, Oct 2006, pp. 130–134.
[15] C. J. Fung, O. Baysal, J. Zhang, I. Aib, and R. Boutaba, "Trust management for host-based collaborative intrusion detection," in *Managing Large-Scale Service Deployment*, F. De Turck, W. Kellerer, and G. Kormentzas, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 109–122.
[16] W. Li, Y. Meng, and L. Kwok, "Enhancing trust evaluation using intrusion sensitivity in collaborative intrusion detection networks: Feasibility and challenges," in *2013 Ninth International Conference on Computational Intelligence and Security*, Dec 2013, pp. 518–522.
[17] G. Zyskind, O. Nathan, and A. Pentland, "Enigma: Decentralized computation platform with guaranteed privacy," 06 2015.
[18] N. Alexopoulos, E. Vasilomanolakis, N. Rka Ivnk, and M. Mhlhuser, *Towards Blockchain-Based Collaborative Intrusion Detection Systems: 12th International Conference, CRITIS 2017, Lucca, Italy, October 8-13, 2017, Revised Selected Papers*, 09 2018, pp. 107–118.
[19] W. Meng, E. W. Tischhauser, Q. Wang, Y. Wang, and J. Han, "When intrusion detection meets blockchain technology: A review," *IEEE Access*, vol. 6, pp. 10 179–10 188, 2018.
[20] T. Golomb, Y. Mirsky, and Y. Elovici, "Ciota: Collaborative iot anomaly detection via blockchain," 03 2018.
[21] M. Steichen, S. Hommes, and R. State, "Chainguard a firewall for blockchain applications using sdn with openflow," in *2017 Principles, Systems and Applications of IP Telecommunications (IPTComm)*, Sep. 2017, pp. 1–8.
[22] P. K. Sharma, S. Singh, Y. Jeong, and J. H. Park, "Distblocknet: A distributed blockchains-based secure sdn architecture for iot networks," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 78–85, Sep. 2017.
[23] (2018, Jun) Snort. Https://www.snort.org/.
[24] (2019) Penetration testing software, pen testing security. [Online]. Available: https://www.metasploit.com/